

Department of Mathematics and Statistics, University of Helsinki  
Numerical methods and the C language, Winter and Spring 2016

Workshop 9

Exercise 9 FILE: h09.tex printed on November 20, 2015 at 11:07.

1. The entries of the  $n \times n$  Hilbert matrix  $H$  are  $h_{ij} = 1/(i + j - 1)$ ,  $i, j = 1, \dots, n$ . Consider solving the equation  $Hx = b$  where  $b = H(1, \dots, 1)^T$  for  $n = 5, \dots, 20$  using two methods: (a) LUsolve (b) SVDsolve2 with a suitable epsedit. Compare the accuracy of each method.
2. Fitting a LSQ line  $y = kx + b$  through a prescribed point  $(s, t)$  to a data set  $(x_i, y_i)$ ,  $i = 1, \dots, m$ , has  $k = \sum_{i=1}^m ((x_i - s)(y_i - t)) / \sum_{i=1}^m (x_i - s)^2$ ,  $b = t - ks$ .

(a) Verify these formulas for  $k$  and  $b$ .

Next suppose that we wish to fit a broken line with a break point  $(s, t)$ . Then we will consider the sum of squares

$$g(s, t) = \sum_{i=1; x_i \leq s} (y_i - (k_1 * x_i + b_1))^2 + \sum_{i=1; x_i > s} (y_i - (k_2 * x_i + b_2))^2$$

where  $k_i, b_i$  are given by the formula above, and the summation for  $k_1(k_2)$  is taken over indices with  $x_i$  less (larger) than  $s$ . Finally, we minimize the function  $g(s, t)$ .

- (b) The program `mypwlf1t2.cpp` on the [www](#)-page executes this idea. Generate your own data set for the program and check that the program works correctly.
- (c) Fit the usual LSQ line to the same data and compare the results.
3. The program `mycxint8.cpp` on the [www](#)-page shows how to integrate complex valued functions along a given polygonal path with three simple methods: (i) Riemann sum (ii) trapez formula (iii) Simpson's rule.
  - (a) Use each of these methods to integrate `myfun2` along a polygonal path that goes in the positive direction twice around the origin.
  - (b) Also use each method to compute the line integral of `myfun1` from  $(1,0)$  to  $(3,2)$  along two different paths, each consisting of segments parallel to the coordinate axes.
4. Solve the following systems of equations using the Newton method.

a) With initial values  $x_1 = 2, x_2 = 0$ :

$$\begin{cases} 2(x_1 + x_2)^2 + (x_1 - x_2)^2 - 8 & = 0 \\ 5x_1^2 + (x_2 - 3)^2 - 9 & = 0 \end{cases} .$$

b) With initial values  $x_1 = 3, x_2 = 4, x_3 = 5$ :

$$\begin{cases} 3x_1 - \cos(x_2 x_3) - 0.5 & = 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 10.6 & = 0 \\ \exp(-x_1 x_2) + 20x_3 + (10\pi - 3)/3 & = 0 \end{cases} .$$

5. Modify the program `mycal.cpp` to accept complex numbers as parameters. You should implement at least product, sum and power of the complex numbers.